Sciences Methods and Technologies International Journal (SciMeTech)

ISSN: 3085-5284, journal homepage: www.scimetech.com

Evaluating SciBERT and Large Language Models for Joint Scientific Entity and Relation Extraction

Abdellah EL OMARI, Jilali ANTARI

Laboratory of Computer Systems Engineering-Mathematics and Applications (ISIMA), Polydisciplinary Faculty of Taroudant, Ibn Zohr University, 80000, Agadir, Morocco.

abdellah.elomari.19@edu.uiz.ac.ma, j.antari@uiz.ac.ma

How to cite:

Abdellah EL OMARI, Jilali ANTARI, "Evaluating SciBERT and Large Language Models for Joint Scientific Entity and Relation Extraction", Sciences Methods and Technologies International Journal (SciMeTech), Vol 2, Issue 2, p 1-6

Keywords:

Named entity recognition
Relation extraction
SciBERT
Large Language Models

Abstract

The automatic extraction of structured knowledge from scientific publications re-quires both robust recognition of entities and accurate identification of semantic relations. Traditional domain-specific models such as SciBERT have shown promising results, while recent large language models (LLMs) raise questions about their adaptability to specialized scientific tasks. In this study, we perform a comparative evaluation of joint entity and relation extraction using SciBERT, LLaMA-2, GPT-3.5-turbo, and GPT-4o-mini on the SciERC corpus, which contains 450 annotated scientific abstracts. Experiments are conducted on annotated scientific corpora covering four entity types (Material, Method, Metric, Task) and two relation types (USED-FOR, EVALUATE-FOR). According to the results, SciBERT is still the best model for relation extraction (F1 = 0.674), whereas GPT-4o-mini is best for named entity recognition (F1 = 0.623). In particular, LLaMA-2 and GPT-3.5- turbo show little to no success. This shows a contrast: for relation extraction, domain-specific models outperform general purpose large language models (LLMs), and for named entity recognition, the reverse is true. We suggest mixed approaches, which combine the best of both worlds, as the way forward for building scientific knowledge graphs.

I. Introduction

Because scientific literature is growing so quickly, there is a greater need for automated knowledge extraction methods that turn research results into structured, machine-readable formats [1], [2]. Named Entity Recognition (NER) [3] finds scientific entities like Material, Method, Metric, and Task. Relation Extraction (RE) [4] finds the semantic connections between these entities. These tasks are the basic parts of scientific knowledge graphs. They help with things like finding information and answering questions.

In traditional pipeline methods, NER and RE are seen as separate steps. This causes errors to spread and makes it hard to model how entities and relations are related to each other [5]. Recent joint extraction frameworks mitigate this limitation by concurrently learning entities and relations. Also, sentence-level models work well for local dependencies, but they don't work well for cross-sentence relations. Document-level approaches, on the other hand, are better at modeling long-range dependencies across whole abstracts or articles. This is shown by datasets like SCIREX [6].

This study investigates document-level joint extraction from annotated abstracts by employing a closed schema featuring predefined entity types (*Material*, *Method*, *Metric*, *Task*) and relation types (*USED-FOR*, *EVALUATE-FOR*), which are deliberately designed to capture and emphasize the methodological structure and dependencies underlying scientific research contributions. We compare SciBERT [7], a pretrained model for a specific field, with fine-tuned large language models (LLMs) like LLaMA-2 [8], GPT-3.5-turbo, and GPT-4o-mini [9]. The experimental evaluation reveals a distinct trade-off between these model families: SciBERT excels in relation extraction, while GPT-4o-mini outperforms

in entity recognition. These results show the strengths of both domain-specific and general-purpose models and encourage the use of hybrid methods for extracting scientific information.

II. Methods

Our methodology is designed to evaluate and compare domain-specific and large language models for document-level joint entity and relation extraction from scientific texts. This section details the dataset, entity and relation schema, and the models under comparison.

a. Dataset

We conduct our experiments on the SciERC [10] corpus, which contains 450 annotated scientific abstracts from the field of artificial intelligence. Each abstract is annotated with four predefined entity types (Material, Method, Metric, Task) and two relation types (USED-FOR, EVALUATE-FOR).

For training and evaluation, we adopt the standard split of the corpus: 350 abstracts for training and 100 for validation. Table 1 summarizes the distribution of entities and relations across the two subsets.

Table 1. Distribution of entities and relations in the SciERC dataset.

Split	Docs	Material	Method	Metric	Task	USED-FOR	EVALUATE-FOR
Train	350	542	1445	231	894	1690	313
Val	100	158	430	71	263	533	91

b. Models Compared

To investigate the trade-off between domain-specialized and general-purpose architectures, we evaluate the following models:

- SciBERT: a BERT-based transformer pretrained on 1.14M scientific papers from Semantic Scholar, providing a strong domain-specific baseline. We fine-tune Sci-BERT for joint NER and RE.
- LLaMA-2-7B: a large language model adapted using parameter-efficient fine-tuning (LoRA). Prompts are designed to enforce structured outputs (JSON format) for entity and relation prediction.
- GPT-3.5-turbo: fine-tuned on our dataset with a constrained schema for entity and relation extraction.
- GPT-4o-mini: a lightweight GPT-4 variant, fine-tuned with structured prompting to ensure consistent extraction.

This combination allows us to contrast domain-specialized and general-purpose LLMs, shedding light on their complementary strengths.

c. Experimental Setup

i. SciBERT

We fine-tuned SciBERT, a BERT-based transformer that had been trained on a large collection of scientific papers from Semantic Scholar, to be our baseline for this domain. SciBERT gives you token-level contextual representations that fit well with scientific language.

We adapted SciBERT for joint named entity recognition and relation extraction by adding two task-specific heads to the pretrained encoder. The entity recognition head is a token classification layer predicting the four predefined entity types (*Material*, *Method*, *Metric*, *Task*) plus the non-entity label. The relation extraction head builds span-pair representations from the encoder outputs to classify each candidate pair into one of the two predefined relation types (*USED-FOR*, *EVALUATE-FOR*) or "no relation." Both heads share the encoder parameters and are trained simultaneously, which encourages the model to exploit interdependencies between entity boundaries and relation cues.

Before fine-tuning, all abstracts from the training split of SciERC were tokenized and entity spans aligned with the annotated schema. Input sequences were truncated or segmented to respect the model's maximum token length (256 tokens) while preserving inter-sentence context.

Fine-tuning used AdamW (lr = 1.85×10^{-5}, batch = 16) for 7 epochs, with 0.2 dropout and a linear warm-up over the first 10% of training steps. **Figure 1** illustrates the overall SciBERT-based joint entity and relation extraction model.

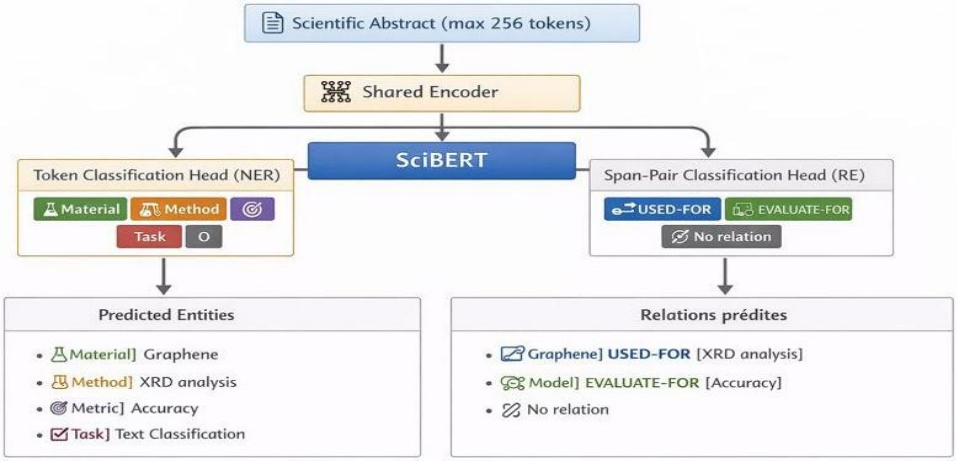

Fig. 1. Joint Entity and Relation Extraction with SciBERT

Formally, the model minimizes a joint training objective combining the losses of both tasks:

$$L = L_{\text{NER}} + \lambda L_{\text{RE}}$$

where L_{NER} and L_{RE} are the negative log-likelihood losses for named entity recognition and relation extraction respectively, and λ controls the relative weight of the two components.

Figure 2 below illustrates the evolution of training and validation losses for NER and RE based on SciBert.

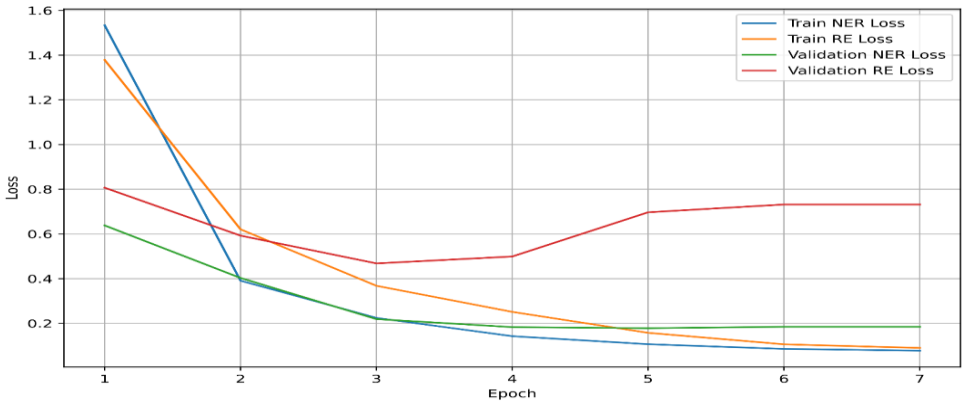

Fig. 2. SciBERT — Training vs. Validation Loss (NER and RE)

ii. LLaMA-2

We adapt LLaMA-2-7B to joint entity and relation extraction using a parameter-efficient fine-tuning setup (QLoRA/LoRA) and instruction-style supervision over document-level abstracts. In contrast to the encoder-based SciBERT setup, LLaMA-2 is trained as a conditional generator: given a formatted prompt (instructions + abstract), the model must produce a strict JSON that contains both entities and relations under a closed ontology (entities: *Material*, *Method*, *Metric*, *Task*; relations: *USED-FOR*, *EVALUATE-FOR*).

Dataset & prompt format. Training examples are provided in JSONL (one example per line). Each item contains the abstract text and the gold structured target. We convert them to instruction-tuning pairs as follows:

- Input (prompt) – system/user template that (i) states the task and the closed schema, (ii) injects the abstract, (iii) reminds the output format.
- Target (completion) – gold JSON with exact fields and labels.

Gold targets strictly follow a predefined schema: entities are annotated with unique IDs, character-level spans, and types from a closed set, while relations link head–tail entity IDs using labels from a closed set when applicable. **Figure 3** illustrates the prompted training setup for LLaMA-2, from instruction/schema and abstract concatenation to the generation of a strictly structured JSON target.

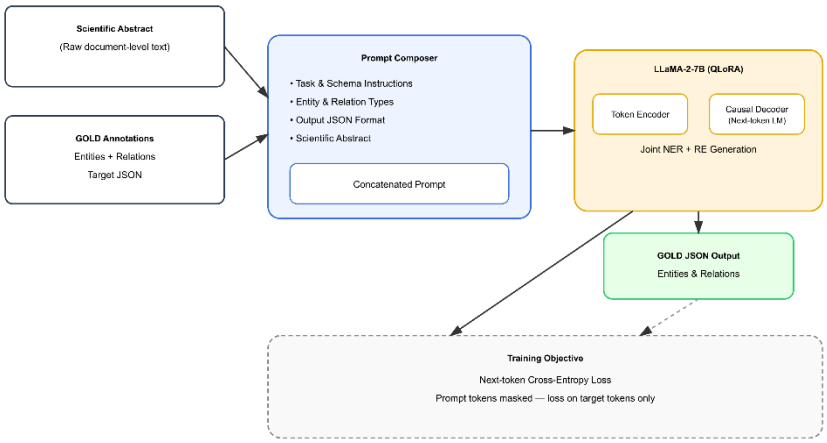

Fig. 3. Prompted Training Setup for LLaMA-2 (Joint NER + RE)

Model adaptation (QLoRA/LoRA). We fine-tune only low-rank adapters on attention/MLP projections while keeping the base weights frozen, which reduces memory and preserves generalization. Formally, for a linear layer with frozen base weights $W_0 \in R^{d_{\text{out}} \times d_{\text{in}}}$, LoRA applies a low-rank update:

$$W = W_0 + \frac{\alpha}{r} BA, \qquad \boldsymbol{A} \in R^{r \times d_{\text{in}}}, \boldsymbol{B} \in R^{d_{\text{out}} \times r}.$$

where r is the adapter rank and α a scaling factor. We apply this to selected attention and MLP projections under QLoRA.

Preprocessing. Abstracts are normalized (whitespace, Unicode) and truncated to 1024 tokens (MAX_SEQ_LEN), budgeting for the instruction/schema prompt, the abstract, and the structured JSON target; targets are enforced to be JSON-only and schema-conformant via a lightweight sanitizer.

iii. GPT-3.5-turbo and GPT-4o-mini

We fine-tuned GPT-3.5-turbo-1106 and GPT-4o-mini-2024-07-18 on the OpenAI platform for document-level joint entity and relation extraction under a closed ontology (entities: Material, Method, Metric, Task; relations: USED-FOR, EVALUATE-FOR). **Figure 4** summarizes the supervised fine-tuning pipeline used in our experiments.
Training uses chat-style JSONL examples: the prompt states the task, schema, and abstract; the target is a strict JSON containing typed entity spans with ids and relations referencing head/tail ids.
We fine-tuned both models with the same hyperparameters—*n_epochs* = 3, *batch_size* = 1, and *learning_rate_multiplier* = 1.8—to ensure a comparable training regime. Under this setup, the training footprint totaled 740,502 trained tokens for GPT-3.5-turbo-1106 and 733,674 trained tokens for GPT-4o-mini. Both models are optimized as conditional generators to emit the full structured output; temperature is kept at 0 for sanity checks to promote determinism and adherence to the closed schema.

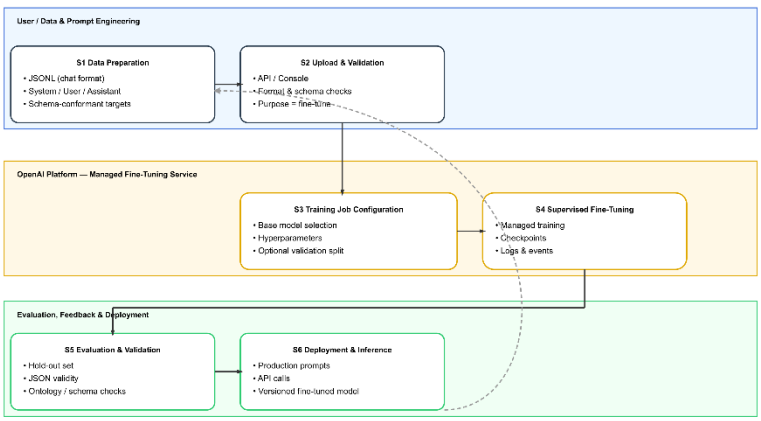

Fig. 4. Supervised Fine-Tuning Pipeline on the OpenAI Platform

Figure 5 below shows the training and validation losses over training steps for the GPT-4o-mini model.

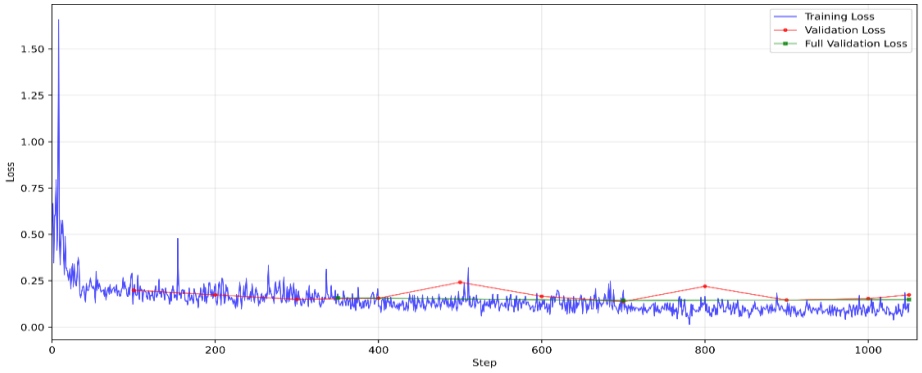

Fig. 5. GPT-4o-mini — Training and validation Loss over Steps

III. Results and discussion

Here we report results on the SciERC corpus (train = 350, val = 100), using exact-span metrics for NER and exact (head, tail, type) matching for RE. We compare SciBERT, LLaMA-2-7B, GPT-3.5-turbo, and GPT-4o-mini under a unified evaluation protocol (Precision/Recall/F1).

a. NER extraction Results

Table 2 reports exact-span NER performance on SciERC. GPT-4o-mini achieves the best F1 score (0.623), exhibiting a well-balanced precision–recall trade-off (0.626/0.619). Compared to the strongest encoder-based baseline, SciBERT, it yields a substantial +0.060 absolute F1 improvement (+10.7%). Other models lag behind, highlighting the advantage of instruction-tuned LLMs for this task.

Table 2. NER results (exact-span) on SciERC: Precision, Recall, and F1 for all models.

Model	Precision	Recall	F1
SciBERT	0.542	0.585	0.563
LLaMA-2-7B	0.475	0.599	0.530
GPT-3.5-turbo	0.513	0.491	0.502
GPT-4o-mini	**0.626**	**0.619**	**0.623**

The models show different precision-recall patterns, even though they all have the same overall ranking. LLaMA-2 tends to have high recall but low precision, which could mean that it is over-detecting or drifting boundaries. SciBERT is more balanced but still leans toward recall. GPT-4o-mini has almost the same precision and recall, which means that span localization is consistent under the closed schema. GPT-3.5-turbo is pretty accurate, but its recall is a little lower. These patterns support our larger point: GPT-4o-mini is great at NER, but SciBERT, which is specialized for a certain field, does better later in relation extraction (following section).

b. Relation extraction results

SciBERT is clearly better than the other models at extracting relations. It has the highest F1 score and a balanced precision-recall profile. GPT-4o-mini, GPT-3.5-turbo, and LLaMA-2-7B, on the other hand, are behind. They either have low precision (spurious links) or lower recall (missed cross-sentence relations). These trends are a result of the modeling choices: The domain-specialized encoder and discriminative span-pair head of SciBERT do a good job of capturing document-level, schema-constrained relations. On the other hand, generative LLMs are more likely to have schema drift and link relations too much or too little.

Table 1. Relation extraction on SciERC (exact head–tail–type matching; P/R/F1).

Model	Precision	Recall	F1
SciBERT	**0.6321**	**0.7671**	**0.6742**
LLaMA-2-7B	0.041	0.114	0.060
GPT-3.5-turbo	0.098	0.067	0.079
GPT-4o-mini	0.242	0.222	0.231

IV. Conclusion

We compared document-level joint NER+RE on SciERC under a closed ontology using SciBERT, LLaMA-2-7B, GPT-3.5-turbo, and GPT-4o-mini. A clear trade-off emerged: GPT-4o-mini was strongest for NER, while SciBERT dominated RE. Generative LLMs, trained to emit structured JSON, were competitive for span detection but more error-prone for relation linking; conversely, the domain-specialized, discriminative setup of SciBERT better captured schema-constrained relations.

These results suggest a hybrid path: leverage LLMs for high-quality entity spans, then apply a SciBERT-style discriminative head for reliable relation decisions. Main limitations include the size of SciERC (450 abstracts), and the restricted ontology (two relation types). Future work will focus on hybrid training setups in which LLM-generated entity spans inform discriminative relation extraction, and on extending the ontology and domains to test generalization.

References

[1] Z. Hong, L. Ward, K. Chard, B. Blaiszik, and I. Foster, "Challenges and advances in information extraction from scientific literature: a review," JOM, vol. 73, no. 11, pp. 3383–3400, 2021.

[2] M. H. A. Abdullah, N. Aziz, S. J. Abdulkadir, H. S. A. Alhussian, and N. Talpur, "Systematic literature review of information extraction from textual data: recent methods, applications, trends, and challenges," IEEE Access, vol. 11, pp. 10535–10562, 2023.

[3] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recogni-tion," IEEE Trans. Knowl. Data Eng., vol. 34, no. 1, pp. 50–70, 2020.

[4] K. Detroja, C. K. Bhensdadia, and B. S. Bhatt, "A survey on relation extraction," Intelli-gent Systems with Applications, vol. 19, p. 200244, 2023.

[5] Z. Yan, Z. Jia, and K. Tu, "An empirical study of pipeline vs. joint approaches to entity and relation extraction," in Proceedings of the 2nd Conference of the Asia-Pacific Chap-ter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2022, pp. 437–443.

[6] S. Jain, M. Van Zuylen, H. Hajishirzi, and I. Beltagy, "SciREX: A challenge dataset for document-level information extraction," arXiv preprint arXiv:2005.00512, 2020.

[7] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," arXiv preprint arXiv:1903.10676, 2019.

[8] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv pre-print arXiv:2307.09288, 2023.

[9] K. Abramski, S. Citraro, L. Lombardi, G. Rossetti, and M. Stella, "Cognitive network science reveals bias in GPT-3, GPT-3.5 Turbo, and GPT-4 mirroring math anxiety in high-school students," Big Data and Cognitive Computing, vol. 7, no. 3, p. 124, 2023.

[10] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," arXiv preprint arXiv:1808.09602, 2018.